

Introducing Development with Pig

In this video, we cover the essentials of Pig, as well as several of its commands.

Apache Pig is a scripting platform for analyzing large data sets that consists of a high level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turn enables them to handle very large data sets. It consists of a compiler that produces sequences of MapReduce programs or Tez, for which large-scale parallel implementations already exist. Pigs Language Layer currently consist of a textual language called Pig Latin which has the following key properties.

Ease of programming,

it is trivial to achieve parallel execution of simple, embarrassingly parallel data analysis tasks.

Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand and maintain optimization opportunities.

The way in which tasks are encoded permits the system to optimize their execution automatically allowing the user to focus on semantics rather than efficiency.

Extensibility, users can create their own functions to do special purpose processing The main use cases are ETL, iterative data processing, and research on raw data.

Let us take a look at the major components.

The language for

the platform is called Pig Latin, which abstracts from the Java map reduce idiom into a form similar to SQL.

While SQL is designed to query the data, Pig Latin allows you to write a data flow that describes how your data will be transformed such as aggregate, join and sort.

Since Pig Latin Scripts can be graphs instead of requiring a single output, it is possible to build complex data flows involving multiple inputs, transforms and outputs.

The compiler compiles the optimized logical plan into a series of MapReduce jobs.

It converts Pig Latin to an optimized MR or Tez program.

Execution engine, finally the map reduce jobs are submitted to Hadoop in a sorted order,

the engine in which Pig Latin commands are executed.

Then, these MapReduce jobs are executed on Hadoop producing

the desired result.

It supports both local and Hadoop modes.

Now let us take a look at the major components.

Apache Pig supports both simple data types such as int, long, float, double, chararray, bytearray and boolean.

And complex types including map, tuple and bag.

The data model of Pig Latin is fully nested and it allows complex, non-atomic data types such as map and tuple.

A record that is formed by an ordered set of fields is known as a tuple.

The fields can be of any type.

A tuple is similar to a row in a table of a relational database.

A bag is an ordered set of tuples.

In other words,

a collection of tuples being non-unique is known as a bag.

Each tuple can have any number of fields providing a flexible schema.

A map or data map is a set of key value pairs.

The key needs to be of type chararray and should be unique.

The value might be in any type.

Here are various commands in Pig Latin.

You can use the load command to read data from the file system.

Write data to the file system using the store command.

You can use dump to write output to stdouts.

And with foreach, apply expressions to each record and generate one or more records.

The filter command applies predicates to each record, and removes the record where false.

Group Cogroup, collects records with the same key from one or more inputs.

And Join is used to join two or more inputs based on a key.

Order sorts records based on a key, and distinct removes duplicate records.

Union merges two data sets.

Use limit to reduce the number of records.

And split is used to split data into two or more sets based on filter conditions.

Here is an example customer data file containing customer details in Pig.

And this is a sample orders data file containing the order details in Pig.

Use the load command to load data in Pig.

This load statement is made up of two parts.

The part to the left of equals is called the relation or alias.

It looks like a variable but

you should note that this is not a variable.

When this statement is executed, no MapReduce task is executed. Since our data set has records with fields separated by pipe, we use the key word using PigStorage with a pipe in parenthesis.

You can also see that some of the data is separated by a comma.

In those cases, we employ the USING PigStorage(',') keyword.

Another thing we've done in the statement here is to give the names of the fields using the as keyword.

To query data, use the following keywords, filter.

Pig allows you to remove unwanted records based on a condition.

This filter functionally is similar to the where clause in SQL.

The filter operator in Pig is used to remove unwanted records from the data file.

The syntax of the filter operator is shown on the slide.

Join, you can use join as SQL to join the data.

Dump is to display the data on screen and to store the output to the HDFS, use store.